CLAIMS

What is claimed is:

1. An apparatus for retaining maximum speed of a flip-flop metastability based random number generator, comprising:

a fixed delay unit having an input for receiving a common signal from a digital signal generator, said fixed delay unit providing a fixed period of delay to the signal as an output;

a variable delay unit having an input for receiving the common signal from the digital signal generator, said variable delay unit being tunable to provide a variable delay to the common signal as an output;

a pair of NAND gates each of which has a first input that receives a respective output of one of fixed delay unit and variable delay unit; an output of a first NAND gate is input to a second NAND gate of the pair of NAND gates, and an output of the second NAND gate is input to the first NAND gate of the pair of gates;

a frequency measurement and delay tuning module that receives an output of a first NAND gate of the pair of NAND Gates, said module checks the frequency of random number bit generation and updates variable delay unit to according to predetermined criteria to tune the variable delay unit so as to maximize the speed of the random bit generation.

2. The apparatus according to claim 1, wherein the frequency measurement made by the frequency measurement and delay tuning module occurs at predetermined intervals.

3. The apparatus according to claim 2, wherein the frequency measurement and delay tuning module comprises a microprocessor.

4. The apparatus according to claim 2, wherein the frequency measurement and delay tuning module comprises a counter that is incremented each time a random bit is produced, and multiplied by a weight (<1) at every clock cycle and according to the following algorithm:

```
#define Weight 0.9990234375 //1 - 1/(2<<10)
Counter = Counter * Weight + IsRandomBitGenerated();
```

5. The apparatus according to claim 2, wherein the predetermined delay schedule is determined by an algorithm that determines whether the variable delay should be equal, smaller, or larger than a present value.

6. The apparatus according to claim 5, wherein the algorithm comprises the following MATLAB code:

```
qlen = 100;                                  % Queue length
dmax = 256;                                  % #delay values
dsig = sqrt(dmax);                           % standard deviation of steps
speed = zeros(1,qlen);                       % start with speed 0
delay = zeros(1,qlen) + dmax/2;              % start with median delay
i = 1;                                       % insertion point in the queue
while 1                                      % infinite loop to keep max speed
  dstep = randn*dsig;                        % steps of normal distribution
  dstep = sign(dstep)*ceil(abs(dstep));      % ensure |step| > 0
  [smx,imx] = max(speed);                    % last max and its index in queue
  dmx = delay(imx);
  dly = max(1,min(256,dmx + dstep));         % next try
  spd = GetSpeed(dly);                       % set delay, get speed
  delay(i) = dly;                            % store trial results
  speed (i) = spd;
  i = i + 1;                                 % move insertion point in queue
  if i > qlen, i = 1; end
end.
```

7. The apparatus according to claim 1, wherein delay values and corresponding speed measured by frequency measurement and delay module during a last N number of occurrences are stored in a queue.

8. The apparatus according to claim 1, wherein delay values and corresponding speed measured by frequency measurement and delay module during a last N number of occurrences are stored in a dynamic table.

9. The apparatus according to claim 7, wherein the delay values are a pseudo-random sequence of values having a Gaussian distribution.

10. A computer readable medium comprising the following algorithm of executable instructions for generating random numbers:

(i) setting a queue length at a predetermined value;

(ii) setting a predetermined number of delay values;

(iii) designating a standard deviation (dmax) of steps;

(iv) starting with speed of 0;

(v) starting with a median delay;

(vi) setting an insertion point in the queue while keeping an infinite loop at maximum speed;

(vii) designating a number of steps of normal distribution;

(viii) ensuring that |step| > 0;

(ix) obtaining a last maximum speed and its index in the queue;

(x) setting the delay as imax;

(xi) repeating for next next delay value (from 1 to 256);

(xii) setting delay (dly) and getting speed (spd);

(xiii) storing trial results of speed and updating a variable delay unit used for random number generation;

(xiv) moving/increasing insertion point i in the queue by 1;

(xv) if the insertion point i > que length, and i = 1, then ending the routine;

(xvi) go to step (xi).

11. A system for retaining maximum speed of flip-flop metastability based random number generation comprising:

means to receive a common signal from an output of at least one flip-flip;

a fixed delay unit having an input for receiving the common signal from the flip-flop, said fixed delay unit delaying the output of the signal by a fixed period before being output;

a variable delay unit having an input for receiving the common signal from the flip-flop, said variable delay unit being tunable to provide a variable delay to the common signal as an output;

a pair of NAND gates, each of which has a first input that receives a respective output of one of fixed delay unit and variable delay unit; an output of a first NAND gate is input to a second NAND gate of the pair of NAND gates, and an output of the second NAND gate is input to the first NAND gate of the pair of gates;

a frequency measurement and delay tuning module that receives an output of a first NAND gate of the pair of NAND Gates, said module checks the frequency of random number bit generation and updates variable delay unit to according to predetermined criteria to tune the delay so as to maximize the speed of the random bit generation.

12. The system according to claim 11, wherein the frequency measurement made by the frequency measurement and delay tuning module occurs at predetermined intervals.

13. The system according to claim 11, wherein the frequency measurement and delay tuning module comprises a microprocessor.

14. The system according to claim 11, wherein the frequency measurement and delay tuning module performs the following tuning algorithm:

(i) setting a queue length at a predetermined value;

(ii) setting a predetermined number of delay values;

(iii) designating a standard deviation (dmax) of steps;

(iv) starting with speed of 0;

(v) starting with a median delay;

(vi) setting an insertion point in the queue while keeping an infinite loop at maximum speed;

(vii) designating a number of steps of normal distribution;

(viii) ensuring that $|step| > 0$;

(ix) obtaining a last maximum speed and its index in the queue;

(x) setting the delay as imax;

(xi) repeating for next next delay value (from 1 to 256);

(xii) setting delay (dly) and getting speed (spd);

(xiii) storing trial results of speed and updating a variable delay unit used for random number generation;

(xiv) moving/increasing insertion point i in the queue by 1;

(xv) if the insertion point i > que length, and i = 1, then ending the routine;

(xvi) go to step (xi).

15. A method for retaining the maximum speed of a flip-flop metastability based random number generator comprising the following steps:

(a) measuring the frequency of random bit generation;

(b) determining whether a predetermined period of time has passed since a previous adjustment of a variable delay unit;

(c) if the predetermined period in step (b) has passed, determining whether the frequency measured in step (a) is at a maximum;

(d) if it has been determined in step (c) that the frequency is at a maximum, reverting to step (a), otherwise, adjusting the variable unit by a predetermined amount according to an algorithm that determines whether the variable delay should be larger, equal to, or smaller than optimum frequency by a predetermined amount based on the frequency measured in step (a);

(e) resetting a timer that measures a predetermined period of time between adjustments of the variable unit, and returning to step (a).

16. The method according to claim 15, wherein the frequency measured in step (a) is performed by a counter and by using the following C code:

```
#define Weight 0.9990234375 //1 - 1/(2<<10)
Counter = Counter * Weight + IsRandomBitGenerated ();.
```

17. The method according to claim 15, wherein the determination of maximum frequency in step (c) is made by comparing values in a table.

18. The method according to claim 15, wherein the determination of maximum frequency is step (c) is made by correlating values in a ring-buffer.

19. The method according to claim 15, wherein the algorithm that is used to adjust the variable unit in step (d) comprises the following steps :

(i) setting a queue length at a predetermined value;

(ii) setting a predetermined number of delay values;

(iii) designating a standard deviation (dmax) of steps;

(iv) starting with speed of 0;

(v) starting with a median delay;

(vi) setting an insertion point in the queue while keeping an infinite loop at maximum speed;

(vii) designating a number of steps of normal distribution;

(viii) ensuring that |step| > 0;

(ix) obtaining a last maximum speed and its index in the queue;

(x) setting the delay as imax;

(xi) repeating for next next delay value (from 1 to 256);

(xii) setting delay (dly) and getting speed (spd);

(xiii) (xiii)storing trial results of speed and updating a variable delay unit used for random number generation;

(xiv) moving/increasing insertion point i in the queue by 1;

(xv) if the insertion point i > que length, and i = 1, then ending the routine;

(xvi) go to step (xi).